REMARKS

Original claims 1-28 have been examined. Applicants acknowledge and appreciate the Examiner's indication that claims 6-9 and 20-23 define allowable subject matter. The remaining claims 1-5, 10-19, and 24-28 are rejected as being unpatentable over "Introduction To Algorithm" by Cormen, Leiserson and Rrivest (herein "CLR") in view of "Indexing Large Metric Spaces for Similarity Search Queries" by Bozkaya and Tolga (herein "BT").

An interview was held on April 4, 2003 between Grace Law (Applicants' representative) and Examiner Marcin R. Filipczyk. The discussion focused primarily on the application of the CLR and BT references to claims 1 and 15, but no agreement was reached during the interview. During the interview, it was noted to the Examiner that the median of the BT reference refers to a median of distance of the first vantage point and the second vantage point. Examiner Filipczyk indicated further consideration was needed in view of Applicants' comments, and requested that Applicants submit their comments in writing. This response is submitted pursuant to that request.

APPLICANTS' RESPONSE

Section 112 rejection, Second Paragraph – Claims 5, 8, 13, 14, 19, 22, 27, and 28

Claims 5, 8, 13, 14, 19, 22, 27, and 28 are rejected under 35 U.S.C. §112, second paragraph, as being indefinite. In response, Applicants amend claims 13 and 27 to correct the minor informalities suggested by the Office action. Moreover, claims 1, 11, 15, and 25 are similarly amended to clarify the features of the present invention. However, Applicants traverse the Office action's objection with respect to claims 5, 8, 19, and 22.

5, 8, 19 & 22

In regard to claims 5 and 19, Applicants note that the term "wrapper" is known in the art. Specifically, a wrapper is defined as code that is combined with another piece of code to determine how that other code is executed, and the wrapper generally acts as an interface between its caller and the wrapped code (See "Attachment A"). Thus, Applicants submit that the use of the term "wrapper," as recited in the claims, is proper.

In regard to claims 8 and 22, the claims recite that the features of the descendent nodes of the binary tree include data that represent the number of times one or more threads of execution have passed through one or more code modules. The data are included in the descendent nodes

of the binary tree, and the data are tracked from the list of elements. In light of Applicants' explanation, Applicants submit that claims 8 and 22 need not be amended. Accordingly, for all these reasons, Applicants request that the Section 112, first paragraph, rejection of claims 5, 8, 13, 14, 19, 22, 27, and 28 be withdrawn.

Section 103 rejection – Claims 1-5, 10-19, and 24-28

In an effort to expedite prosecution, Applicants herein amend independent claims 1, 11, and 13 to include the allowable subject matter of cancelled claim 6 and independent claims 15, 25, and 27 to include allowable subject matter of cancelled claim 20. Moreover, dependent claims 7, 8, and 9 and dependent claims 21, 22, and 23, which have been indicated as definitely allowable subject matter, are amended into independent form to include features of claims 1 and 15, respectively. Since claims 2, 12, 14, 16, 26, and 28 depend from independent claims 1, 11, 13, 15, 25, and 27, they are patentable for at least the reasons set forth above with regard to their corresponding independent claims. Similarly, dependent claims 3-5, 10, 17-19, and 24 are patentable for at least the reasons set forth above with regard to independent claims 1 and 15, from which they respectively depend. Applicants reserve the right to present further arguments in the future with regard to the corresponding claims and the dependent claims in the event that the independent claims are found to be unpatentable. Accordingly, Applicants request that the Section 103 rejection of claims 1-5, 10-19, and 24-28 be withdrawn.

New Claims – Claims 29-66

Moreover, Applicants have added claims 29-66 to include the allowable subject matter of claims 7-8 and 21-23. In particular, Applicants added (1) dependent claims 29-33, 34-38, and 39-43, which depend from claims 7, 8, and 9, respectively, corresponding to dependent claims 3-5 and 10; (2) dependent claims 48-52, 53-57, and 58-62, which depend from claims 21, 22, and 23, respectively, corresponding to dependent claims 17-19 and 24; (3) independent claim 44 including the features of claim 11 and allowable subject matter of claim 7; (4) independent claim 46 including the features of claim 13 and allowable subject matter of claim 7; (5) independent claim 63 including the features of claim 25 and allowable subject matter of claim 21; (6) independent claim 65 including the features of claim 27 and allowable subject matter of claim

44 = 7+11
46 = 13+7
63 = 25+21
65 = 27+21

21; and (7) dependent claims 45, 47, 64, and 66 depending from independent claims 44, 46, 63, and 65, respectively. Accordingly, Applicants respectfully request the Examiner's consideration and allowance of these new claims.

In view of the foregoing amendments and remarks, Applicants submit that the present application is in condition for allowance. An early and favorable action is earnestly requested.

CONCLUSION

The application is considered in good and proper form for allowance, and the Examiner is respectfully requested to pass this application to issue. If, in the opinion of the Examiner, a telephone conference would expedite the prosecution of the subject application, the Examiner is invited to call the undersigned attorney.

Respectfully submitted,

Grace Law, Reg. No. 48,872
One of the Attorneys for Applicant(s)
LEYDIG, VOIT & MAYER, LTD.
Two Prudential Plaza, Suite 4900
180 North Stetson
Chicago, Illinois 60601-6780
(312) 616-5600 (telephone)
(312) 616-5700 (facsimile)

Date: July 7, 2003

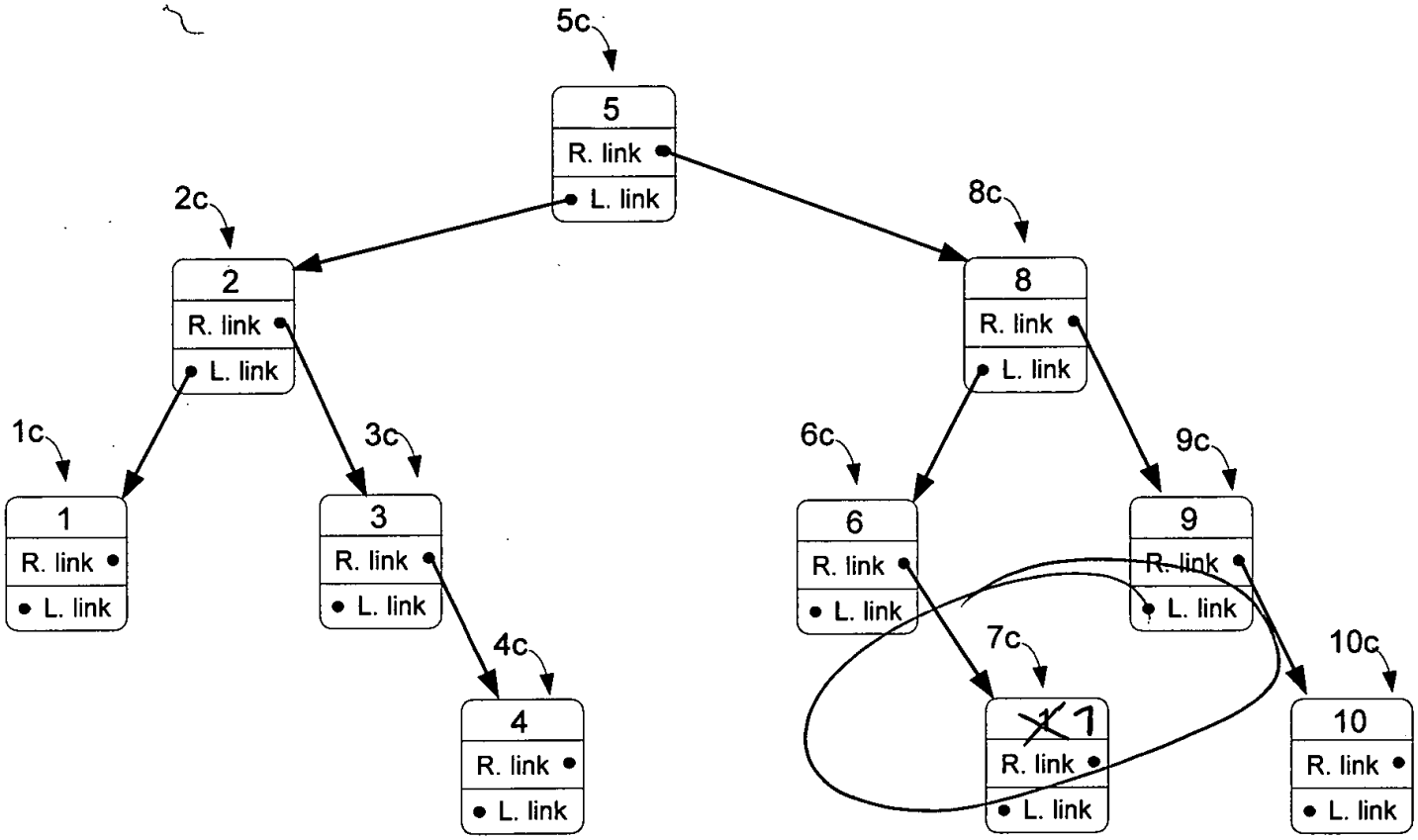

FIG. 11

wrapper

<*programming*> Code which is combined with another piece of code to determine how that code is executed. The wrapper acts as an interface between its caller and the wrapped code. This may be done for compatibility, e.g. if the wrapped code is in a different programming language or uses different calling conventions, or for security, e.g. to prevent the calling program from executing certain functions. The implication is that the wrapped code can only be accessed via the wrapper.

(1998-12-15)

Try this search on OneLook / Google

Nearby terms: wps « WRAM « wrap around « **wrapper** » wrats nest » wrb » write

---

"ATTACHMENT A"